

Revisiting Bounded Suboptimal Heuristic Search

Jordan Thayer

University of New Hampshire
Durham NH 03824 U.S.A
Department of Computer Science
jtd7@cs.unh.edu

Abstract

A* is the optimal algorithm for finding optimal solutions to heuristic search problems. Given the same amount of information, no search can expand fewer nodes. Many applications of heuristic search only require that we find solutions of reasonable quality or in a reasonable amount of time, where reasonable is defined by the needs of a user. My doctoral work will address these problems by developing new bounded suboptimal algorithms which perform better than the previous approaches. I will show that anytime searches can incorporate these new algorithms to improve their own performance. I will demonstrate that anytime search is not the correct approach when a deadline is known at the beginning of the search, and introduce deadline-aware search algorithms that address this setting directly.

1 The Problem

Heuristic search is used to solve a variety of problems, such as pathfinding in a grid or temporal planning. If sufficient resources are available, optimal solutions can be found using A* search with an admissible heuristic [Hart, Nilsson, & Raphael, 1968]. Finding optimal solutions can require a large amount of time and memory. This can be unacceptable, and so we must explore techniques for limiting the time and memory consumed while finding a solution.

In practical settings we must accept suboptimal solutions in order to reduce the computation required to find solutions. If we want control when abandoning optimality, we can either bound the quality of the solution returned by a search or we can bound the amount of time used to find a solution. When bounding the quality of a solution our aim then becomes to solve problems as quickly as possible while guaranteeing that the solutions found have a cost within a bounded factor of optimal. When we choose to bound the time, our goal is to produce the best possible solution within the deadline.

2 Research Plan

First, we consider the setting in which one wants the fastest possible search where the solution is guaranteed to be within

a bounded factor of the optimal solution. For a given factor w , we call an algorithm w -admissible if it is guaranteed to return a solution that is no more than w times more expensive than an optimal solution.

Weighted A* is an elegant answer to this problem [Pohl, 1970]. In weighted A* the traditional node evaluation function of A*, $f_{A^*}(n) = g(n) + h(n)$, is modified to place additional emphasis on the heuristic evaluation function h , as in $f_{wA^*}(n) = g(n) + w \cdot h(n)$. The additional focus on h encourages greedy behavior. All solutions found by weighted A* are w -admissible.

Weighted A* finds solutions of a given quality quickly, but it is not necessarily the fastest algorithm. The dissertation will present three approaches for improving weighted A* and bounded suboptimal search in general: using additional information, improving the heuristic online, and exploiting the average case performance of existing algorithms.

Additional Information The proof that A* is the optimal algorithm for finding optimal solutions is based on the assumption that all algorithms are equally informed [Dechter & Pearl, 1988]. Giving an algorithm additional information, typically in the form of multiple heuristics can improve its performance, even beyond that of A* in some domains.

Improving Heuristics Heuristics used in an algorithm can be improved by looking at their accuracy over the course of the search. This provides a measure of the error present in the heuristic allowing us to correct the function online using various learning techniques such as temporal difference learning and incremental linear regression.

Average Case Performance The proofs guaranteeing solutions returned by a search are of a certain quality focus on the worst case, where the solution found has a quality exactly equal to the bound. These algorithms often return solutions which are well within the bound as opposed to right at the bound. If we assume that the solutions returned by a heuristic search algorithm are better than the bound the suggests, we can improve search performance by running a fast search which is not guaranteed to return a solution within the desired bound, and then showing that this solution is within the bound later.

Second, we consider the problem of finding solutions un-

der a deadline. Here, the user is constrained by time and not by cost, though ideally we'd like to find the best solution possible within the deadline. An algorithm only returns the best solution found within the time allotted, and not returning a solution within the deadline is unacceptable. I will address problems where the deadline is unknown as well as problems where the deadline is set in advance.

Anytime heuristic search [Hansen & Zhou, 2007] and Anytime repairing A* [Likhachev, Gordon, & Thrun, 2004] are a natural fit for problems where the deadline is unknown. They produce costly solutions quickly improving the solution as time allows. Both of these algorithms rely on weighted A*, which may not be the best bounded suboptimal algorithm for all domains. Our improved bounded suboptimal search algorithms into anytime algorithms following a technique outlined by Hansen & Zhou [2007].

Finally, I will address the problem of search under a specified deadline. One flaw in the anytime approach to this problem is that only the final solution is returned, so all computation spent on finding any solution other than the one returned is, in a sense, wasted. Despite this obvious problem, anytime search is still the most widely used technique for solving problems under a deadline. Building predictive models of the time it takes to find a solution of a particular quality given a search configuration will allow for a direct addressing of the stated problem. With predictive models in hand, a search may more wisely invest the time up to the deadline allowing it to find solutions of a higher quality within the same time bound.

3 Progress To Date

I have been primarily addressing search under a suboptimality bound. With collaborators, I have looked at deadline search and have a preliminary approach to this problem. As it relies on the performance of an underlying bounded suboptimal search I have been focusing my attention there. We have discovered three general techniques for improving bounded suboptimal search and used them to produce new search algorithms which are better than the current state of the art in several domains.

Optimistic search [Thayer & Ruml, 2008] takes advantage of the good average case performance of weighted A*. It searches using a weight higher than the desired suboptimality bound, assuming that this will find a solution within the desired bound very quickly. Then it expands additional nodes in order to show that the solution in hand is within the desired bound. It takes advantage of weighted A*'s tendency to return solutions that are much better than the bound suggests. It performs well in domains where weighted A* performs well so long as they don't have a large number of duplicate states.

Revised dynamically weighted A*, an extension of dynamically weighted A* [Pohl, 1973], takes advantage of a second heuristic function. Unlike approaches that combine several cost-to-go estimates [Samadi, Felner, & Schaeffer, 2008], this algorithm considers the search distance-to-go. It uses this distance estimate to decrease the weight applied to the heuristic as nodes become closer to a goal state. Weighting implicitly rewards nodes near solutions, as they tend to have low heuristic values. Lowering the weight as the distance to go

estimate decreases explicitly rewards progress. Revised dynamically weighted A* works best where solution cost and solution length differ. It can find optimal solutions faster than A* because the additional heuristic is used for tie-breaking.

Clamped adaptive search [Thayer, Ruml, & Bitton, 2008] attempts to improve the quality of the heuristic evaluation function online. Currently, it uses a simple global error model that assumes error is evenly distributed across the search space. Error is measured by comparing then f values of a parent node and its best child. We can then use any number of features and machine learning techniques to calculate an improved heuristic. This algorithm performs well for tight bounds in pathfinding for grids and a few other domains. It does not become greedy as the bound becomes generous which leads to decreased performance at high bounds. We are attempting to address this problem by taking the heuristic correction of clamped adaptive and using it to improve the behavior of A_ϵ^* [Pearl & Kim, 1982]. This has shown great promise in pathfinding which warrants further study.

References

- [Dechter & Pearl, 1988] Dechter, R., and Pearl, J. 1988. The optimality of A*. In Kanal, L., and Kumar, V., eds., *Search in Artificial Intelligence*. Springer-Verlag. 166–199.
- [Hansen & Zhou, 2007] Hansen, E. A., and Zhou, R. 2007. Anytime heuristic search. *JAIR* 28:267–297.
- [Hart, Nilsson, & Raphael, 1968] Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems Science and Cybernetics* SSC-4(2):100–107.
- [Likhachev, Gordon, & Thrun, 2004] Likhachev, M.; Gordon, G.; and Thrun, S. 2004. ARA*: Anytime A* with provable bounds on sub-optimality. In *Proceedings of NIPS 16*.
- [Pearl & Kim, 1982] Pearl, J., and Kim, J. H. 1982. Studies in semi-admissible heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4(4):391–399.
- [Pohl, 1970] Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial Intelligence* 1:193–204.
- [Pohl, 1973] Pohl, I. 1973. The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computation issues in heuristic problem solving. In *Proceedings of IJCAI-73*, 12–17.
- [Samadi, Felner, & Schaeffer, 2008] Samadi, M.; Felner, A.; and Schaeffer, J. 2008. Learning from multiple heuristics. In *Proceedings of AAAI-08*.
- [Thayer & Ruml, 2008] Thayer, J. T., and Ruml, W. 2008. Faster than weighted A*: An optimistic approach to bounded suboptimal search. In *Proceedings of ICAPS-2008*.
- [Thayer, Ruml, & Bitton, 2008] Thayer, J. T.; Ruml, W.; and Bitton, E. 2008. Fast and loose in bounded suboptimal heuristic search. In *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*.